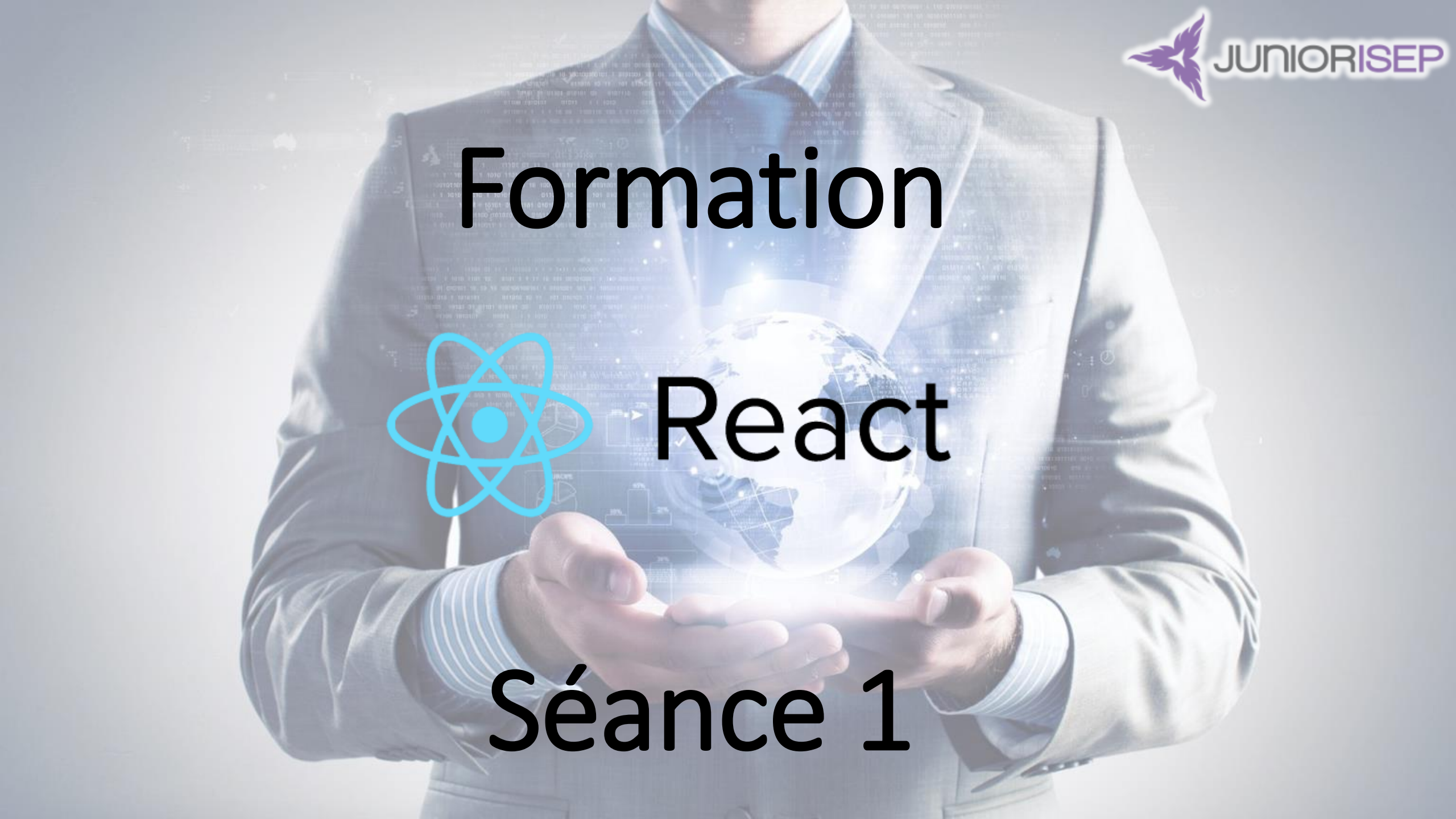

JUNIORISEP
Formation
React
Séance 1

Installation

- Node.js -> https://nodejs.org/en/download/
- Npm
- Create-React-App -> npm install --global create-react-app

React

- Bibliothèque front-end open-source gérée par Facebook (peut être considéré comme un Framework)
- Approche par composants
- Interface utilisateur très performante
- Utilisé par Airbnb, Discord, Facebook, Twitter …

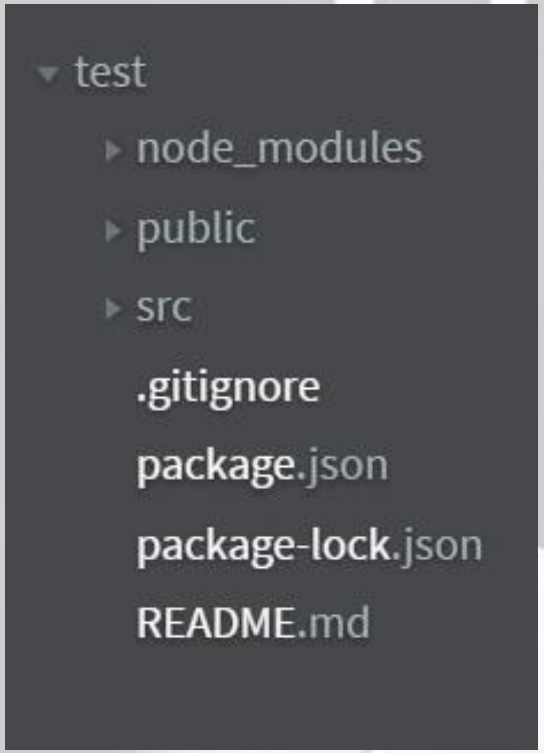
Création du projet

- create-react-app <nom du dossier>
- npm start

Structure d’un projet

Fichiers requis :

- public/index.html
- src/index.js



Import

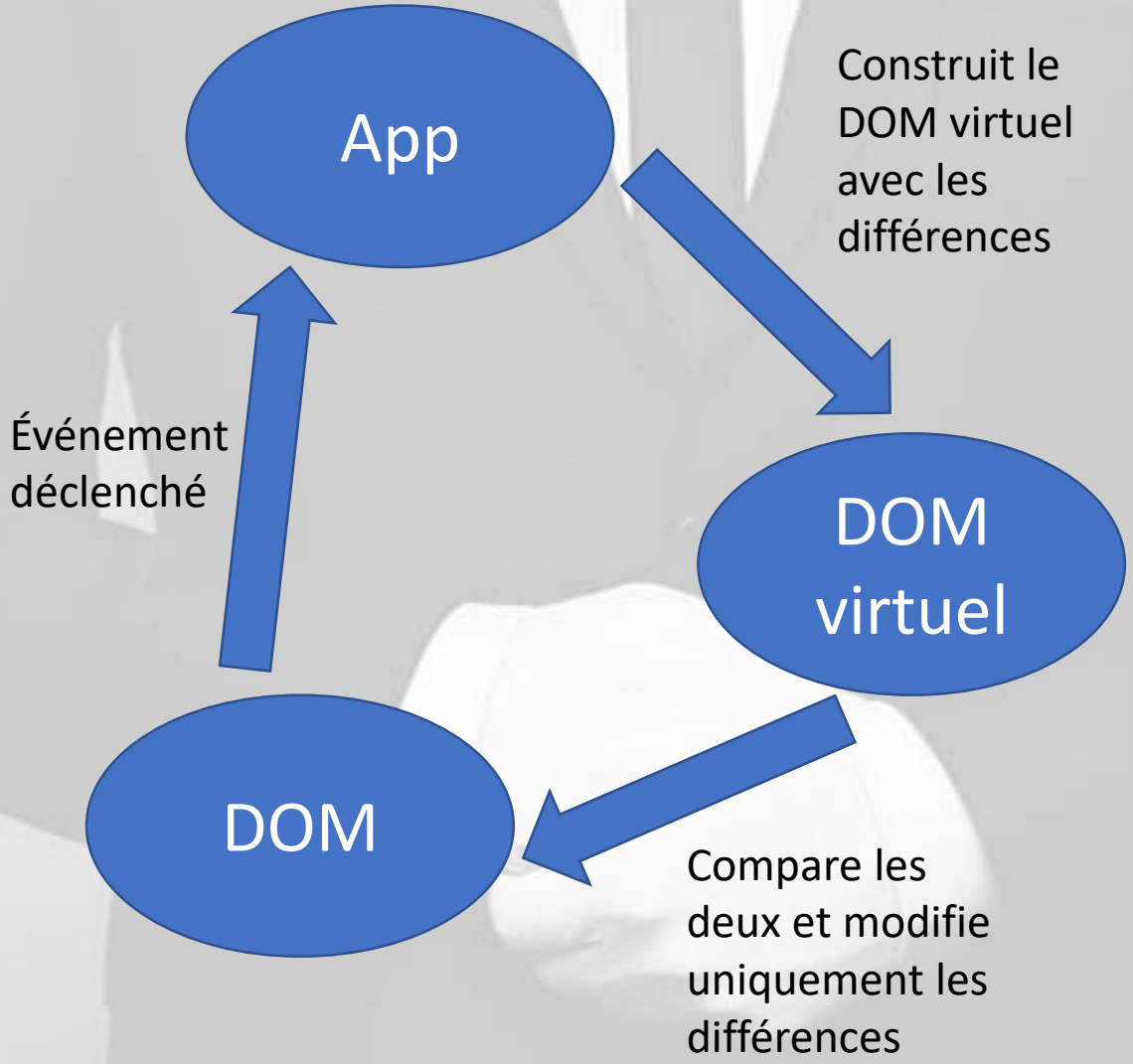
- import React (fonctionnalités react)
- import ReactDOM (DOM virtuel)
- import App (composant)
- Import {Component} (pour définir un composant)

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
```

```
import React, { Component } from 'react';
```

DOM virtuel

- Le DOM virtuel permet à React de gérer l'interface utilisateur efficacement en modifiant le DOM que partiellement



Composants

- Structure imbriquée
- Composant parent et enfant
- render()
- JSX
- Majuscule au nom

```
class App extends Component {
  render() {
    return (
        <div>
            <p>coucou</p>
        </div>
    );
  }
}
```

```
function Coucou(){
    return <p>coucou</p>;
}
```

JSX

- Sensible à la casse
- for et class sont réservés => htmlFor et className
- Majuscule aux noms des composant, minuscule aux noms des éléments natifs

```
return  <div>
            <h2 className="pinkText">Coucou</h2>
            <p className="redText">Du texte</p>
        </div>

return  React.createElement(
            "div",
            null,
            React.createElement(
              "h2",
              { className: "pinkText" },
              "Coucou"
            ),
            React.createElement(
              "p",
              { className: "redText" },
              "Du texte"
            )
        );
```

Afficher son composant

```
class App extends Component {
  render() {
    return (
        <div>
            <p>coucou</p>
        </div>
    );
  }
}
```

```
ReactDOM.render(<App />, document.getElementById('root'));
```

```
<body>
  <noscript>
    You need to enable JavaScript to run this app.
  </noscript>
  <div id="root"></div>

</body>
```

State

- Etat interne d’un composant
- constructor() défini les états avec this.state
- setState() permet d’actualiser la valeur des états
- onChange appel la méthode handleChange à chaque changement dans l’input

```
constructor(props){
    super(props)
    this.state = {
        contenu:'coucou!'
    }
    this.handleChange = this.handleChange.bind(this)
}
```

```
handleChange(e){
    this.setState({
        contenu: e.target.value
    })
}
```

```
<div>
    <p>Contenu : </p>
    <p>{this.state.contenu}</p>
    <input
        type="text"
        onChange={this.handleChange}
    />
</div>
```

Les événements

- Clavier : onKeyDown, onKeyPress, onKeyUp
- Focus : onFocus, onBlur
- Formulaire : onSubmit, onChange
- Souris : onHover, onClick
- …

Notre composant

```
import React, { Component } from 'react';


class App extends Component {
    constructor(props){
        super(props)
        this.state = {
            contenu:'coucou!'
        }
        this.handleChange = this.handleChange.bind(this)
    }
    handleChange(e){
        this.setState({
            contenu: e.target.value
        })
    }
  render() {
    return (
        <div>
            <p>Contenu : </p>
            <p>{this.state.contenu}</p>
            <input
                type="text"
                onChange={this.handleChange}
            />
        </div>
    );
  }
}

export default App;
```

Props

- Les props sont des informations passées d'un composant parent vers un composant enfant
- Un enfant ne doit jamais modifier directement une props

```
import Compteur from './Compteur';
```

```
<div>
    <p>Contenu : </p>
    <p>"{this.state.contenu}"</p>
    <Compteur chaine={this.state.contenu}/>
    <input
        type="text"
        onChange={this.handleChange}
    />
</div>
```

```
{this.props.chaine}
```

Un peu de style

- Importer le fichier css
- class != className
- Style inline possible

```
<div style={{border:'1px solid black', display:'flex'}}>
```

```
import './App.css';
```

```
<h3
id="superTitre"
className="titre"
>
    Contenu :
</h3>
```

```
#superTitre{
    color:red;
}

.titre{
    font-weight: bold;
}
```

TP

- Objectif: Créer une calculatrice
- Utiliser eval() => plus simple mais peut mener à des erreurs
- Gérer chaque opération indépendamment

Résultat